

This application is filed in the name of the following inventor(s):

<i>Inventor Name</i>	<i>Citizenship</i>	<i>Residence City and State</i>
Stephane KASRIEL	France	Mountain View, California
Walter MANN	United States	San Francisco, California

The assignee is *Fireclick, Inc.*, a corporation having a place of business at 289 South San Antonio Road, Suite 150, Los Altos, California 94022.

Title of the Invention

New Template Identification with Differential Caching

Background of the Invention

1. Field of the Invention

This invention relates to distinguishing different elements in a web page so as to reduce the bandwidth and computing resources required to transmit a web page over a network.

2. *Related Art*

One technique to optimize bandwidth usage and save computing resources involves responding to a request for a web page by looking to what information a client device may have already received from a server at an earlier point in time. As shown in the Incorporated Disclosures, information included in a document may be distinguished from other information included in that same document by looking to how frequently the information changes and whether a user has the most recent version of that part of the document. Using these criteria, information in a document may be categorized as either (1) template information (including information that changes infrequently) or (2) delta information (including information that changes relatively frequently or is somehow personalized with respect to a user). Differential caching and service of template and delta information results in a saving of bandwidth and other computing resources because information that does not change frequently may be compressed and cached at a server that is relatively proximate to one or more end users.

Differential caching based upon the distinction between template information and delta information does not necessarily result in a substantial savings of bandwidth and computing resources under all conditions, particularly under those conditions when amount of delta information exceeds the amount of template information. For example, different elements in a personal web page,

1 such as those provided by Yahoo, may change at very different rates with respect
2 to each other.

3
4 For example, the headline stories presented to a user may change
5 every hour, whereas elements associated with the user's decision to view a
6 particular topic may change only occasionally, and still other elements (such as the
7 title bars in a page or logos associated with the provider of the content) may
8 change very rarely. Under these circumstances, the template contains relatively
9 little information compared to the amount of delta information. Under these
10 circumstances, differential caching may not significantly reduce the amount of
11 bandwidth and other resources needed to transmit a web page because a large part
12 of the page must be freshly obtained from the content delivery network or re-
13 compressed every time it is transmitted.

14
15 Accordingly, it would be desirable to provide a technique for
16 efficiently serving relatively non-static content in a content delivery network that
17 does not suffer from the drawbacks of the known art.

18 19 Summary of the Invention

20
21 In a first aspect of the invention, information in a document is
22 categorized with respect to the number of times that the information is served to a

1 client or a set of clients within a period of time. The information in a document
2 may be categorized as (1) delta information (that is, information that changes
3 rapidly or is unique to a particular user), (2) sub-template information (including
4 information that changes less frequently than delta information, and (3) template
5 information, which changes very seldom. For example, a personalized web page
6 (such as those available through Yahoo) may include (1) template information,
7 such as the Yahoo logo and associated headers; (2) one or more sets of sub-
8 template information, such as news reports, televisions listings and comparable
9 information that change at more or less regular intervals, and (3) delta information,
10 such as personal messages to an individual, stock market quotes or other relatively
11 ephemeral messages.

12
13 In a second aspect of the invention, both the template information
14 and sub-template information are compressed and cached. Compressing and
15 caching both sub-template information and template information results in a
16 significant savings of bandwidth and computing resources, such as would be
17 required if the sub-template information were treated as delta information and
18 were not stored in a cache as is the case in the prior art. This savings is enhanced
19 when the compressed template and sub-template information are sent to a large
20 number of users.

1 In a preferred embodiment, the sub-template and template
2 information are treated separately, so that each is compressed and cached in a
3 different location or a different memory address from the other. When a user
4 requests a document, the document may be comprised of template, sub-template
5 and delta information, some of which the requesting user or other users may have
6 seen before. Selective compression and caching of these elements with respect to
7 the likelihood that these elements will change can result in a savings of
8 bandwidth and computing resources that is not available in the prior art.

9
10 In a third aspect of the invention, the compressed and cached
11 versions of the template and sub-template information are tagged with an indicator
12 (referred to herein as an etag) that identifies the version number of a template or
13 sub-template, so as to distinguish it from other, previous versions that a user may
14 have received or that may be cached in a server, either in the content delivery
15 network or on a proxy server. In a preferred embodiment, other types of electronic
16 tags, such as place markers, are used to identify a location in the page where the
17 material is to be inserted and provide other related information.

18
19 In a fourth aspect of the invention, a hierarchy of templates and sub-
20 templates is created, so as to form a lattice structure where a given sub-template
21 can be shared by multiple templates. In a first example, a personalized web page
22 associated with a user may include some of the same sub-templates as a different

1 personalized web page associated with a different user. In this example, a
2 template is associated with each user or group of users; this template references
3 sub-templates which contain portions of the contents of the page (e.g. a sub-
4 template for the weather section of the page, a sub-template for the news section,
5 and so on). Such sub-templates can be shared by several templates (as occurs
6 when several users wish to watch the same news). In a second example, sub-
7 templates can be used in a personalized media stream as when a user watches a
8 news story which includes personalized advertisement. If the number of different
9 personalized advertisement streams is much lower than the number of individual
10 viewers, it is most efficient to treat each advertisement stream as a sub-template,
11 personalization information as delta information and the rest of the data stream as
12 a template.

13
14 In a preferred embodiment, differential caching of template and sub-
15 template information and differential service of template, sub-template and delta
16 information is particularly useful with respect to personalized web pages that are
17 comprised of elements that can be configured in many different ways and sent to
18 many different users. In other embodiments, this technique can be used with
19 regard to database queries, streaming media and other information obtained from a
20 network.

1 In a preferred embodiment, an encoder specifies how the final page
2 is to be assembled, but not where the assembly should occur. Thus, the web page
3 can be assembled, either in whole or in part, by any downstream element that (1)
4 understands encoding and (2) has access to the cached information.

6 BRIEF DESCRIPTIONS OF THE DRAWINGS

8 Figure 1 is an overview of a system for new template identification
9 with differential caching.

11 Figure 2 is an exemplary view of different elements in a web page
12 that are differentially cached.

14 Figures 3A and 3B are flow diagram, illustrating a method for using
15 a system for new template identification with differential caching.

17 DESCRIPTION OF THE PREFERRED EMBODIMENT

19 The invention is described herein with regard to preferred steps and
20 data structures. Those skilled in the art will recognize, after perusal of this
21 application, that the described steps and data structures are not limited to any
22 particular processing devices (whether general-purpose or special-purpose

1 processing devices, or specific circuitry). Rather, those of ordinary skill in the art
2 would be able to implement the described steps and data structures, and
3 equivalents thereof, without undue experimentation or further invention. All such
4 implementations are within the scope and spirit of the invention.

5
6 Inventions described herein can be used in conjunction with
7 inventions described in the following applications:

- 8
- 9 • Application Serial No.09/436,136, filed November 9, 1999, in the name of
10 Stephane KASRIEL, Xavier CASANOVA and Walter MANN, titled
11 “Predictive Pre-Download of Network Objects”, attorney docket number
12 155.1001.02
13
 - 14 • Application Serial No. 09/734,910, filed December 11, 2000, in the name
15 of Stephane KASRIEL, Xavier CASANOVA and Walter MANN, titled
16 “Predictive Pre-Download Using Normalized Network Objects Identifiers”,
17 attorney docket number 155.1002.01
18
 - 19 • Application Serial No.09/827,268, filed April 4, 2001, in the name of
20 inventor Stephane KASRIEL and Walter MANN, titled “Server-Originated
21 Differential Caching”, attorney docket number 155.1003.02
22

- 1 • Application Serial No. 09/888,374, filed June 22, 2001, in the name of
2 Stephane KASRIEL and Walter MANN, titled “ Content Delivery Network
3 Using Differential Caching”, attorney docket number 155.1005.01
4
- 5 • Application Serial No. _____, filed _____, in the name of
6 Stephane KASRIEL and Walter MANN, titled “Efficient and Low-Cost
7 Compression Using Differential Caching” attorney docket number
8 155.1006.01.

9
10 Each of these applications is hereby incorporated by reference as if
11 fully set forth herein. They are collectively referred to as the “incorporated
12 disclosures”.

13
14 *System Elements*

15
16 Figure 1 is an overview of a system for new template identification
17 with differential caching.

18
19 A system for new template identification with differential caching
20 (shown by general character reference 100) includes one or more clients 110, an
21 originating server 120, a content delivery network 130, a set of proxy encoder
22 servers 140 and a communication network 150.

Each of the one or more clients 110 includes a client workstation 111 and a client operator 112.

The client workstation 111 includes a computing device, along with a local memory, operating system software and a display element. In a preferred embodiment, the computing device includes a personal computer, a hand held or laptop computer, a telephone interface (such as a cellular phone) to such a remote computer or any other device that fits the general Turing paradigm.

The client workstation 111 also includes a web browser 113. In some embodiments, the client workstation 111 also includes a decoder 114.

The web browser 113 (such as "Internet Explorer", "Netscape Navigator" or a comparable product) uses a message transfer protocol, such as HTTP (hypertext transfer protocol), or a variant thereof, to generate request messages 115 (not shown) and receive content (for example, web pages or information from a database) or other messages from the content delivery network 130.

The decoder 114 is either a browser add-on or is coupled to a proxy server locally close to the client 110 (for example, an enterprise or ISP cache). Regardless of location, the decoder 114 causes the web browser 113 to interact

1 with the set of proxy encoder servers 140, identifies previous versions of content
2 that have already been received, decompresses compressed content and integrates
3 the various elements comprising the content into a display for presentation to the
4 client operator 112. In a preferred embodiment, the decoder 114 is coupled to the
5 web browser 113, preferably as a browser add-on.

6
7 Other embodiments (known as “clientless”) do not include a decoder
8 114. In these clientless embodiments, the encoder encodes the templates and delta
9 information at a level understandable by the browser 113 using DHTML, XML or
10 some other scripting technique.

11
12 The client operator 112 might include one or more individual
13 persons or a proxy for one or more such persons (for example, a human
14 administrative assistant, or a computer program or other artificial intelligence
15 system acting on behalf of another).

16 17 The Originating Server and Content Delivery Network

18
19 The originating server 120 includes a processor, computer program
20 and data memory, and operates under control of software to perform the tasks
21 described herein. It is capable of using a message transfer protocol, such as HTTP
22 or a variant thereof, to receive request messages 115 for documents (for example,

1 web pages) from clients 110, the content delivery network 120 or the proxy
2 encoding server 140 and to respond to those request messages 115. In a preferred
3 embodiment, the originating server 120 is the original provider of content.

4
5 The content delivery network 130 includes a set of mirroring servers
6 135 situated in different locations throughout the network so as to minimize
7 bandwidth required to respond to request messages 115. Similar to the originating
8 server 120, the content delivery network 130 includes at least one processor,
9 computer program and data memory, operates under control of software and is
10 capable of using a message transfer protocol, such as HTTP or a variant thereof, to
11 receive request messages 115 for documents.

12
13 In a preferred embodiment, both the originating server 120 and the
14 content delivery network 130 each include a database of documents such as web
15 pages, embedded objects for web pages, databases or any information such as may
16 be requested by a client operator 112.

17
18 In alternative embodiments, the content delivery network 130 may
19 also include various databases for caching template information, sub-template
20 information, delta information to be inserted into template or sub-template web
21 pages, and code fragments or compressed versions of any of the above. In both
22 preferred and alternative embodiments, the mirroring servers 135 include most or

1 all of the same content as the originating server 120, but are positioned more
2 locally to the client device 110. Information is served from the mirroring servers
3 135 to the client devices 110 so as to minimize bandwidth used to transmit content
4 to a client device 110. In this way, the mirroring servers 135 act as a unified
5 content delivery network 120.

7 Proxy encoder server

9 Similar to originating servers 120 and mirroring servers 135, the
10 proxy encoder server 140 also includes a processor, computer program and data
11 memory, and operates under control of software to perform the tasks described
12 herein. The proxy encoder server 140 also includes compression software 142 and
13 software 144 for identifying elements in a web page. Compression software 142
14 may include gzip or glib or some other comparable product that performs a
15 compression algorithm such as Huffman coding or arithmetic coding. Software
16 144 for identifying elements in a web page can (1) distinguish, (2) isolate and (3)
17 tag template information, sub-template information and delta information prior to
18 compression, as well as (4) cache the compressed information.

19
20 The proxy encoder server 140 is capable of using a message transfer
21 protocol, such as HTTP or a variant thereof, to receive request messages 115 for
22 documents (such as for example, web pages) from clients 110, the content delivery

1 network 130 or the originating server 120 and to respond to those request
2 messages 115.

3
4 In a preferred embodiment, proxy encoder servers 140 are logically
5 local to the originating server 120. However, in other embodiments, they may be
6 positioned between the client devices 110 and the content delivery network 130.
7 Requests from the client device 110 that are originally directed to the originating
8 server 120 or the content delivery network 130 are redirected to the proxy encoder
9 server 140. In a preferred embodiment, the proxy encoder server 140 is
10 transparent to the client device 110, the originating server and the content delivery
11 network 130.

12 13 Communication Network

14
15 Client devices 110, the originating server 120, the content delivery
16 network 130 and the proxy encoder server 140 are coupled using a communication
17 network 150. In a preferred embodiment, the communication network 150
18 includes a computer communication network, such as the Internet. However, in
19 alternative embodiments, the communication network 150 might include an
20 intranet, extranet, VPN (virtual private network), ATM system, a portion of a
21 private or public PSTN (public switched telephone network), a frame relay

1 system, or any other communication technique capable of performing the
2 functions described herein.

4 Content Distribution

6 In a preferred embodiment, request messages 115 are generated by
7 the client 110. The request messages 115 are sent directly to the content delivery
8 network 130 or the originating server 120. If there is a proxy encoder 140 “in
9 front” of either the content delivery network 130 or the originating server 120, the
10 request message 115 will be received by the proxy encoder 140. The actions of
11 the proxy encoder 140 are transparent to both the client 110 and the server. The
12 proxy encoder server 140 retrieves the document, including template information
13 and sub-template information (if such information is available) from the content
14 delivery network 130. If template and sub-template information are not available,
15 software 144 is used to dissect web page and break it down into template
16 information, sub-template information or delta information. Each of these
17 components separately compressed, tagged with an etag to provide additional
18 information and cached at the proxy encoder server 120.

20 In an alternative embodiment, each mirroring server 135 maintains a
21 copy of template information and sub-template information for a particular
22 document. When the document is requested by a client 110, the mirroring server

1 135 provides the template information and sub-template information to the client
2 110 from its cache, while obtaining the delta information from the originating
3 server 120 (or from a content distribution network 130 similarly disposed for
4 distributing delta information). Sending the template information and sub-
5 template information from the originating server 120 to the mirroring servers 135
6 is separate from sending the delta information from the originating server 120 to
7 the client 110.

8
9 In a preferred embodiment, partial assembly of the template and sub-
10 template information may occur at the mirroring server 135 or any other server
11 that understands encoding and has access to the cached information. In this
12 embodiment, the client 110 does not have to make multiple requests to the
13 mirroring server 135 for different information. This is particularly beneficial to
14 the client 110 who generally requires the total document, rather than the individual
15 sub-templates.

16
17 Figure 2 is an exemplary view of different elements in a document
18 that can be differentially cached.

19
20 The document (indicated by general character reference 200)
21 includes template information 210 (also referred to as a “template”), sub-template

1 information 220 (also referred to as a “sub-template”) and delta information 230
2 (also referred to as a “delta”).
3

4 Template information 210 (shown in the document 200 as a web
5 page identifier) includes information that is relatively static and does not change
6 frequently. For example, template information 210 may include logos associated
7 with the site provider, title bars that identify the type of information that follows, a
8 list of stocks in a user’s portfolio for which the user frequently requests market
9 quotes and other information that does not change frequently.

10 In a preferred embodiment, the template information 210 is
11 embedded with place markers 212 (not shown). Place markers 212 are used to
12 define a location for the insertion of sub-template information 220 and delta
13 information 230.
14

15 In a preferred embodiment, after the document 200 is requested by a
16 client device 110, the template information 210 is compressed and cached in an
17 address associated with at least one location, such as the proxy encoder server 140
18 or a mirroring server 135. An etag 214 (not shown) identifies the version number
19 of the template information 210.
20

21 Sub-template information 220 includes information that changes
22 relatively frequently, such as a breaking news story, television listings that remain

1 relatively constant for a day, an unchanging weather report and other comparable
2 features. It is not uncommon for sub-template information 220 to be inserted in
3 different locations in a web page during the course of a day. Similarly, different
4 users may display identical sub-template information 220 in different places in
5 their personal web pages. Under these conditions, the sub-template information
6 220 remains unchanged except for its location in a page. As noted above, place
7 markers 212 are used to position such material in a document 200.

8
9 Similar to template information 210, sub-template information 220 is
10 compressed and cached at least one of a number of possible locations, including
11 the mirroring server 135 and the proxy encoder server 140. An etag 214 identifies
12 the version number of the sub-template information 220.

13
14 Delta information 230 includes relatively ephemeral information
15 such as stock quotes, personalized reminders to a client operator 112, advertising
16 content (such as banner ads), weather reports and similar matter that changes
17 frequently.

18
19 In a preferred embodiment, delta information 230 is not compressed
20 or cached, but served directly to the client device 110 either by way of the proxy
21 encoder 140 or directly from the content delivery network 120. In other

embodiments, the delta information 230 may be compressed and cached if it will be reused at a future point in time (such as rotating banner advertisements).

Place markers 212 identify the location where delta information 230 may be inserted into the template information 210. For example, if template information 110 includes a list of stocks for which the client operator 112 requests quotes, the place marker 121 for the delta information (that is the individual stock quotes) will be embedded adjacent to the associated stock names.

In another embodiment, the delta 230 includes code describing where and how to combine the sub-template(s) 220 with the template 210 to generate the document. Such embodiments do not require place markers 121 or 214.

Method of Operation

Figures 3A and 3B are a flow diagram, illustrating a method for using a system for new template identification with differential caching.

A method 300 includes a set of flow points and process steps as described herein.

1 The method 300 is performed by the system 100. Although the
2 method 200 is described serially, the steps of the method 300 can be performed by
3 separate elements in conjunction or parallel, whether asynchronously, in a
4 pipelined manner, or otherwise. There is no particular requirement that the
5 method 300 be performed in the same order in which this description lists the
6 steps, except where so indicated.

7
8 At a flow point 310, the system 100 is ready to begin performing a
9 method 300.

10
11 At a step 311, the client 110 generates a request message 115 for the
12 document 200. The request message 115 can be directed to either the mirroring
13 server 135 or the originating server 120. In a preferred embodiment, the request
14 message 115 is made using the decoder 114, preferably on the client's web
15 browser 113.

16
17 In a step 312, the decoder 114 sends the request message 115 to the
18 proxy encoder 140, situated preferably near the mirroring server 135 or the
19 originating server 120. Since the request message 115 was sent through the
20 decoder 114, the proxy encoder server 140 knows that the client can integrate
21 template information 210, sub-template information 220 and delta information 230
22 and is otherwise compatible with systems that provide delta encoding.

1 In “clientless” versions (that is, those clients without a decoder 114)
2 the request message 115 goes directly from the browser 113 to the proxy encoder
3 server 140, bypassing the decoder 114.

4
5 In a step 313, the proxy encoder server 140 fetches a document 200
6 from either the originating server 120 or the mirroring server 135, depending upon
7 which is closest to the proxy encoder 140. After obtaining this content, the proxy
8 encoder 140 updates the template 210 for the document 200 and compresses the
9 updated template 210. In the event that there is not a template 210 associated with
10 document 200, the proxy encoder 140 generates a template 210, compresses the
11 template 210 and caches it.

12
13 In a step 314, the proxy encoder 140 determines whether sub-
14 templates 220 are desirable and constructs the sub-template(s) 220. This is done
15 by looking to the size of the delta information 230. If the delta 230 is very large
16 with respect to the template 210 and includes regularized information, that
17 regularized information is isolated and used as a sub-template 220. As with the
18 template information 210, the proxy encoder 140 compresses and caches the sub-
19 template information 220.

20
21 In “clientless” embodiments, the proxy encoder 140 responds
22 directly with the delta information 230. In such embodiments, the delta

1 information 230 is an HTML page that includes a reference to template
2 information 210, one or more sets of sub-template information and delta
3 information 230 (wherein the template 210, the sub-template 220 are Javascripts
4 and the delta information includes a set of Javascript instructions). In this
5 embodiment, the Javascript instructions comprising the delta information 230 tell
6 the browser 113 how to transform the template information 210 and sub-template
7 information 220 into the correct HTML document. Other embodiments may use
8 DHTML or other scripting techniques; still others may express the delta
9 information as an XML page or a WML document.

10
11 In a step 315, the decoder 114 retrieves the compressed template 210
12 and the compressed sub-template 220 from either the proxy encoder server 140 or
13 the mirroring server 130, (depending where the encoder proxy 140 specified the
14 compressed template 210 and compressed sub-template were cached in the
15 previous step) by making a request, either immediately or at a later point in time.

16 Upon retrieving the compressed template 210 and compressed sub-template
17 information 220, the decoder 114 decompresses them, and inserts the sub-template
18 information as directly by the place markers 212.

19
20 In the clientless version, the browser 113 automatically and
21 immediately retrieves the template 210 and the sub-template 220 from the site
22 specified in the previous step.

1 In a step 316, the encoder proxy 140 sends the contents of the
2 template 210 and sub-template 220, and one or more etags 214 that correspond to
3 the versions of the template 210 and the sub-template 220.

4
5 If the contents and etag 214 are sent from the mirroring server 130,
6 then the mirroring server 130 searches its cache for the template 110 and sub-
7 template 220. If the template 210 and sub-template 220 are present in its cache,
8 the mirroring server 130 sends the template 210 and the sub-template 220 directly
9 to the client 110. However, if either or both of these elements are not present in
10 the cache, then the mirroring server 130 automatically fetches the missing element
11 from the encoder proxy 140, caches the fetched element, and sends the element to
12 the client 110.

13
14 The following steps occur when a client operator 112 (either the
15 same client operator or a different one) subsequently requests the document 200.

16
17 In a step 317, the client 110 requests a document 200 by generating a
18 request message 115. Similar to steps 310 and 311, the decoder 114 directs the
19 request to the proxy encoder 140, changing as to further specify a version number
20 that is used to ascertain if changes have occurred.

1 In a step 318, the proxy encoder 140 receives the request from the
2 decoder 114. If the proxy encoder 140 determines that a version of the template
3 210 or sub-template 220 are not present in the database, the proxy encoder 140
4 obtains the document 200 from either the originating server 120 or the mirroring
5 server 135, identifies template information 210 and sub-template information,
6 compresses the template 210 and sub-template 220 and caches them.

7
8 If, however, the template 210 and the sub-template 220 are available,
9 the proxy encoder 140 calculates the differences between the versions of template
10 210 and sub-templates 220 that are available in the database and a newer version
11 of the document 200 such as may be available from the originating server 120 or
12 the mirroring server 135. These differences may involve changes in the location
13 where the sub-template is displayed (for example, two different users may choose
14 to have the same information in a personal web page displayed in different
15 locations). Such changes are not true delta information 230 because they can be
16 remedied by positioning information that is already present in a cache differently
17 in a document 200. Other differences may be true delta information 230.

18
19 In a step 319, the proxy encoder 140 may send either the delta
20 information 230 to the decoder 114 (that is, if the decoder 114 can accept delta
21 information 230) or it sends the document 200 to the client 110.

1 Steps 317 through 319 are described with respect to a client – server
2 implementation. In the “clientless” version, the proxy encoder server 140 does
3 not need to know which version of the template 210 and sub-template 220 are at
4 the client 110. This information is not needed because the proxy encoder server
5 140 makes this decision a priori and instructs the client 110 to use a specific
6 version of these elements. Under these circumstances, steps 317 – 318 in the
7 “clientless” version involve the browser 113 (rather than the decoder 114)
8 directing the request message 115 to the encoder proxy 140.

9 10 *Alternative Embodiments*

11
12 Although preferred embodiments are disclosed herein, many
13 variations are possible which remain within the concept, scope and spirit of the
14 invention; these variations would be clear to those skilled in the art after perusal of
15 this application.